MAX PLANCK GESELLSCHAFT

MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

Bridging continuous and discrete tensor representations of multivariate functions via QTT

Bonan Sun (Max Planck Institute Magdeburg)

GAMM Annual Meeting, Mar 19, 2026

Based on a joint work with
Peter Benner (MPI Magdeburg)
Boris Khoromskij (MPI Magdeburg and MPI MiS Leipzig)

Partners:

MAX PLANCK INSTITUTE
FOR MATHEMATICS IN THE SCIENCES

Organization

Discrete and continuous tensor formats

Consider $f : [-1, 1]^D \to \mathbb{R}$, $D = 3$. Approx. f with a *small number of parameters* $\rightsquigarrow$ cheap comput. with f.

1. **Grid-based methods**: discrete Tucker format of function related tensor $\mathbf{F}$ (contains, e.g., function values on a grid):

$$\mathbf{F}(i_1, i_2, i_3) \approx \sum_{j_1=1}^{R} \sum_{j_2=1}^{R} \sum_{j_3=1}^{R} \boldsymbol{\beta}_{j_1,j_2,j_3} \mathbf{u}_{j_1}^{(1)}(i_1) \mathbf{u}_{j_2}^{(2)}(i_2) \mathbf{u}_{j_3}^{(3)}(i_3), \mathbf{u}_{j_\ell}^{(\ell)} \in \mathbb{R}^n, \ \mathbf{F} \in \mathbb{R}^{n\times n\times n}.$$

2. **Mesh-free methods**: functional Tucker format of f:

$$f(x_1, x_2, x_3) \approx f_{\mathbf{m}}(x_1, x_2, x_3) := \sum_{i_1=1}^{R} \sum_{i_2=1}^{R} \sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3) \tag{1}$$

Discrete and continuous tensor formats

Consider $f:[-1,1]^D \to \mathbb{R}$, $D=3$. Approx. f with a *small number of parameters* $\rightsquigarrow$ cheap comput. with f.

1. **Grid-based methods**: **discrete Tucker format** of function related tensor $\mathbf{F}$ (contains, e.g., function values on a grid):

$$\mathbf{F}(i_1,i_2,i_3) \approx \sum_{j_1=1}^{R}\sum_{j_2=1}^{R}\sum_{j_3=1}^{R} \boldsymbol{\beta}_{j_1,j_2,j_3}\mathbf{u}_{j_1}^{(1)}(i_1)\mathbf{u}_{j_2}^{(2)}(i_2)\mathbf{u}_{j_3}^{(3)}(i_3), \mathbf{u}_{j_\ell}^{(\ell)} \in \mathbb{R}^n,\ \mathbf{F}\in\mathbb{R}^{n\times n\times n}.$$

2. **Mesh-free methods**: **functional Tucker format** of f:

$$\begin{aligned} f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) &:= \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1)v_{i_2}^{(2)}(x_2)v_{i_3}^{(3)}(x_3) \\ v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ V^{(\ell)} &\in \mathbb{R}^{m\times R},\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x)). \end{aligned} \tag{1}$$

We call Eq. (1) the **Chebyshev-Tucker (ChebTuck) format**.

ChebTuck to Grid-based tensor

- Consider $f : [-1, 1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1, x_2, x_3) \approx f_{\mathbf{m}}(x_1, x_2, x_3) = \sum_{i_1=1}^{R} \sum_{i_2=1}^{R} \sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$. **Storage**: $\mathcal{O}(DRm + R^D)$.

ChebTuck to Grid-based tensor

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

  with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell)$, $T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$. **Storage**: $\mathcal{O}(DRm + R^D)$.
- **Goal**: Transform ChebTuck format to a discrete tensor format on a fine grid $n \gg m$:

$$\mathbf{F}_{\mathbf{m}}(i_1,i_2,i_3) := f_{\mathbf{m}}(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}), \quad \{t_{i_\ell}^{(\ell)}\}_{i_\ell=1}^{n} \text{ is a uniform grid in } [-1,1].$$

- **Motivation**:
  - Efficient application of discrete operators (differentiation, integration, convolution).
  - Offline-online workflows: expensive offline construction of continuous surrogate, fast online evaluation.
  - High-resolution discretizations for accurate simulations.

ChebTuck to Grid-based tensor

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

  with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$. **Storage**: $\mathcal{O}(DRm + R^D)$.

- **Goal**: Transform ChebTuck format to a discrete tensor format on a fine grid $n \gg m$:

$$\mathbf{F}_{\mathbf{m}}(i_1,i_2,i_3) := f_{\mathbf{m}}(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}), \quad \{t_{i_\ell}^{(\ell)}\}_{i_\ell=1}^{n} \text{ is a uniform grid in } [-1,1].$$

- The **naive approach** yields a discrete Tucker:

$$\mathbf{F}_{\mathbf{m}} = \boldsymbol{\beta} \times_1 U_1 \times_2 U_2 \times_3 U_3, \quad U_\ell(i_\ell, j_\ell) = v_{j_\ell}^{(\ell)}(t_{i_\ell}^{(\ell)})$$

  i.e., each column of U_ℓ contains the discretization of a Chebyshev poly. $v_{j_\ell}^{(\ell)}(x_\ell)$ on the grid $t_{i_\ell}^{(\ell)}$.

- **Storage**: $\mathcal{O}(DRn + R^D)$. **Additional storage** compared to ChebTuck: $\mathcal{O}(DRn)$, $n \gg m$.

ChebTuck to Grid-based tensor

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

  with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$. **Storage**: $\mathcal{O}(DRm + R^D)$.
- **Goal**: Transform ChebTuck format to a discrete tensor format on a fine grid $n \gg m$:

$$\mathbf{F}_{\mathbf{m}}(i_1,i_2,i_3) := f_{\mathbf{m}}(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}), \quad \{t_{i_\ell}^{(\ell)}\}_{i_\ell=1}^{n} \text{ is a uniform grid in } [-1,1].$$

- The **naive approach** yields a discrete Tucker:

$$\mathbf{F}_{\mathbf{m}} = \boldsymbol{\beta} \times_1 U_1 \times_2 U_2 \times_3 U_3, \quad U_\ell(i_\ell, j_\ell) = v_{j_\ell}^{(\ell)}(t_{i_\ell}^{(\ell)})$$

  i.e., each column of U_ℓ contains the discretization of a Chebyshev poly. $v_{j_\ell}^{(\ell)}(x_\ell)$ on the grid $t_{i_\ell}^{(\ell)}$.
- **Storage**: $\mathcal{O}(DRn + R^D)$. **Additional storage** compared to ChebTuck: $\mathcal{O}(DRn)$, $n \gg m$.
- **Remedy:** store columns of U_ℓ as quantized tensor trains (QTT) [Khoromskij '11].

ChebTuck to Grid-based tensor

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

  with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$. **Storage**: $\mathcal{O}(DRm + R^D)$.
- The **naive approach** yields a discrete Tucker:

$$\mathbf{F}_{\mathbf{m}} = \boldsymbol{\beta} \times_1 U_1 \times_2 U_2 \times_3 U_3, \quad U_\ell(i_\ell, j_\ell) = v_{j_\ell}^{(\ell)}(t_{i_\ell}^{(\ell)})$$

  i.e., each column of U_ℓ contains the discretization of a Chebyshev poly. $v_{j_\ell}^{(\ell)}(x_\ell)$ on the grid $t_{i_\ell}^{(\ell)}$.
- **Storage**: $\mathcal{O}(DRn + R^D)$. **Additional storage** compared to ChebTuck: $\mathcal{O}(DRn)$, $n \gg m$.
- **Remedy:** store columns of U_ℓ as quantized tensor trains (QTT) [Khoromskij '11].
- **QTT**: a vector $\mathbf{u} \in \mathbb{R}^n$ with $n = 2^d$, reshape $\mathbf{u}$ to a d-dimensional tensor $\mathbf{U} \in \mathbb{R}^{2\times\cdots\times 2}$ and approximate $\mathbf{U}$ in TT format: $\mathbf{U}(i_1,\ldots,i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d)$ for $i_1,\ldots,i_d = 0,1$.
  Storage: $r^2 d = \mathcal{O}(r^2 \log n)$, where r is the TT rank.

ChebTuck to Grid-based tensor

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

  with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$. **Storage**: $\mathcal{O}(DRm + R^D)$.

- The **naive approach** yields a discrete Tucker:

$$\mathbf{F}_{\mathbf{m}} = \boldsymbol{\beta} \times_1 U_1 \times_2 U_2 \times_3 U_3, \quad U_\ell(i_\ell, j_\ell) = v_{j_\ell}^{(\ell)}(t_{i_\ell}^{(\ell)})$$

  i.e., each column of U_ℓ contains the discretization of a Chebyshev poly. $v_{j_\ell}^{(\ell)}(x_\ell)$ on the grid $t_{i_\ell}^{(\ell)}$.

- **Storage**: $\mathcal{O}(DRn + R^D)$. **Additional storage** compared to ChebTuck: $\mathcal{O}(DRn)$, $n \gg m$.
- **Remedy:** store columns of U_ℓ as quantized tensor trains (QTT) [Khoromskij '11].
- Known: QTT ranks of polynomials p of degree m are bounded by $m+1$ [Khoromskij '11, Oseledets '13] (numerically even $\log m$) $\rightsquigarrow$ $\mathcal{O}(DRn)$ reduced to $\mathcal{O}(DRm^2 \log n)$ (numerically $\mathcal{O}(DR \log^2 m \log n)$).

ChebTuck to Grid-based tensor

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

  with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$. **Storage**: $\mathcal{O}(DRm + R^D)$.
- The **naive approach** yields a discrete Tucker:

$$\mathbf{F}_{\mathbf{m}} = \boldsymbol{\beta} \times_1 U_1 \times_2 U_2 \times_3 U_3, \quad U_\ell(i_\ell, j_\ell) = v_{j_\ell}^{(\ell)}(t_{i_\ell}^{(\ell)})$$

  i.e., each column of U_ℓ contains the discretization of a Chebyshev poly. $v_{j_\ell}^{(\ell)}(x_\ell)$ on the grid $t_{i_\ell}^{(\ell)}$.
- **Storage**: $\mathcal{O}(DRn + R^D)$. **Additional storage** compared to ChebTuck: $\mathcal{O}(DRn)$, $n \gg m$.
- **Remedy:** store columns of U_ℓ as quantized tensor trains (QTT) [Khoromskij '11].
- Known: QTT ranks of polynomials p of degree m are bounded by $m+1$ [Khoromskij '11, Oseledets '13] (numerically even $\log m$) $\rightsquigarrow \mathcal{O}(DRn)$ reduced to $\mathcal{O}(DRm^2 \log n)$ (numerically $\mathcal{O}(DR \log^2 m \log n)$).
- **Fundamental task**: approximate a polynomial p in QTT format efficiently.

From ChebTuck to QTT-Tucker

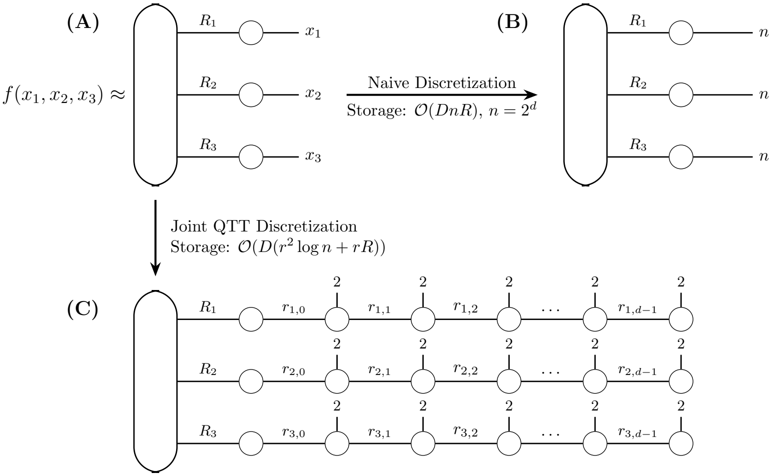


Figure: (A) Continuous ChebTuck. (B) Discrete Tucker ($\mathcal{O}(n)$ storage). (C) QTT-Tucker-like ($\mathcal{O}(\log n)$ storage).

Organization

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + ih\}_{i=0}^{n-1}$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^{n} \in \mathbb{R}^{2^d}$.

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + ih\}_{i=0}^{n-1}$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^{n} \in \mathbb{R}^{2^d}$.
- Use the multi-index mapping $(i_1, \ldots, i_d) \mapsto i = i_1 + 2 \cdot i_2 + \cdots + 2^{d-1} \cdot i_d$ for $i_\ell = 0, 1$
- Reshape $\mathbf{p}$ to a d-dimensional tensor $\mathbf{P} \in \mathbb{R}^{2 \times \cdots \times 2}$:

$$\mathbf{P}(i_1, \ldots, i_d) = p(x_i) = p\Big(- 1 + i_1 h + 2 \cdot i_2 h + \cdots + 2^{d-1} \cdot i_d h\Big)$$

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + ih\}_{i=0}^{n-1}$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^{n} \in \mathbb{R}^{2^d}$.
- Use the multi-index mapping $(i_1, \ldots, i_d) \mapsto i = i_1 + 2 \cdot i_2 + \cdots + 2^{d-1} \cdot i_d$ for $i_\ell = 0, 1$
- Reshape $\mathbf{p}$ to a d-dimensional tensor $\mathbf{P} \in \mathbb{R}^{2 \times \cdots \times 2}$:

$$\mathbf{P}(i_1, \ldots, i_d) = p(x_i) = p\Big(\underbrace{-1 + i_1 h}_{=:x_{i_1}^{(1)}} + \underbrace{2 \cdot i_2 h}_{=:x_{i_2}^{(2)}} + \cdots + \underbrace{2^{d-1} \cdot i_d h}_{=:x_{i_d}^{(d)}}\Big)$$

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + ih\}_{i=0}^{n-1}$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^{n} \in \mathbb{R}^{2^d}$.
- Use the multi-index mapping $(i_1, \ldots, i_d) \mapsto i = i_1 + 2 \cdot i_2 + \cdots + 2^{d-1} \cdot i_d$ for $i_\ell = 0, 1$
- Reshape $\mathbf{p}$ to a d-dimensional tensor $\mathbf{P} \in \mathbb{R}^{2 \times \cdots \times 2}$:

$$
\begin{aligned}
\mathbf{P}(i_1, \ldots, i_d) = p(x_i) &= p\Big(\underbrace{-1 + i_1 h}_{=:x_{i_1}^{(1)}} + \underbrace{2 \cdot i_2 h}_{=:x_{i_2}^{(2)}} + \cdots + \underbrace{2^{d-1} \cdot i_d h}_{=:x_{i_d}^{(d)}}\Big) \\
&= p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}).
\end{aligned}
$$

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + ih\}_{i=0}^{n-1}$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^{n} \in \mathbb{R}^{2^d}$.
- Use the multi-index mapping $(i_1, \ldots, i_d) \mapsto i = i_1 + 2 \cdot i_2 + \cdots + 2^{d-1} \cdot i_d$ for $i_\ell = 0, 1$
- Reshape $\mathbf{p}$ to a d-dimensional tensor $\mathbf{P} \in \mathbb{R}^{2 \times \cdots \times 2}$:

$$\begin{aligned}\mathbf{P}(i_1, \ldots, i_d) = p(x_i) &= p\Big(\underbrace{-1 + i_1 h}_{=: x_{i_1}^{(1)}} + \underbrace{2 \cdot i_2 h}_{=: x_{i_2}^{(2)}} + \cdots + \underbrace{2^{d-1} \cdot i_d h}_{=: x_{i_d}^{(d)}}\Big) \\ &= p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}).\end{aligned}$$

- QTT of p is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2) \cdots \mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 0, 1.$$

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + ih\}_{i=0}^{n-1}$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^{n} \in \mathbb{R}^{2^d}$.
- QTT of p is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 0, 1.$$

Computational methods for the TT of P

1. **Direct method**: apply directly the adaptive TT cross, e.g., `dmrg_cross` [Savostyanov/Oseledets '11].
2. **Constructive method**: there exists an analytic formula[1] for the cores $\mathbf{G}_\ell(i_\ell) \in \mathbb{R}^{(m+1)\times(m+1)}$.

[1] Oseledets, Constructive Approximation 37(1):1–18 (2013)

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + ih\}_{i=0}^{n-1}$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^{n} \in \mathbb{R}^{2^d}$.
- QTT of p is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2) \cdots \mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 0, 1.$$

Computational methods for the TT of P

1. **Direct method**: apply directly the adaptive TT cross, e.g., `dmrg_cross` [Savostyanov/Oseledets '11].
2. **Constructive method**: there exists an analytic formula[1] for the cores $\mathbf{G}_\ell(i_\ell) \in \mathbb{R}^{(m+1)\times(m+1)}$.

Disadvantages: Method 1 does not exploit the 1D nature of the problem, just views $\mathbf{P}$ as a black box tensor. Method 2 quickly becomes numerically unstable & produces pessimistic TT ranks $m + 1$.

[1] Oseledets, Constructive Approximation 37(1):1–18 (2013)

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + ih\}_{i=0}^{n-1}$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^{n} \in \mathbb{R}^{2^d}$.
- QTT of p is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 0, 1.$$

Computational methods for the TT of P

1. **Direct method**: apply directly the adaptive TT cross, e.g., `dmrg_cross` [Savostyanov/Oseledets '11].
2. **Constructive method**: there exists an analytic formula[1] for the cores $\mathbf{G}_\ell(i_\ell) \in \mathbb{R}^{(m+1)\times(m+1)}$.

Disadvantages: Method 1 does not exploit the 1D nature of the problem, just views $\mathbf{P}$ as a black box tensor. Method 2 quickly becomes numerically unstable & produces pessimistic TT ranks $m + 1$.

3. **Our novel method**: Constructive (thus faster than Method 1), stable and rank adaptive.

[1] Oseledets, Constructive Approximation 37(1):1–18 (2013)

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 0, 1.$$

Recall Oseledets' constructive method

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(\underbrace{x_{i_1}^{(1)}}_{=x} + \underbrace{x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}}_{=y}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 0, 1.$$

Recall Oseledets' constructive method

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(\underbrace{x_{i_1}^{(1)}}_{=x} + \underbrace{x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}}_{=y}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 0, 1.$$

Recall Oseledets' constructive method

1. $p(x+y) = \sum_{i=0}^{m} p_i (x+y)^i = \sum_{i=0}^{m} p_i \sum_{\alpha=0}^{i} C_i^\alpha x^\alpha y^{i-\alpha} = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M(\alpha,\beta) x^\alpha y^\beta = \underbrace{[1, x, \ldots, x^m]}_{=X_{\leq m}(x)} M \begin{bmatrix} 1 \\ y \\ \vdots \\ y^m \end{bmatrix}$
   where $M(\alpha,\beta) = p_{\alpha+\beta} C_{\alpha+\beta}^{\alpha}$ for $\alpha + \beta \leq m$ otherwise 0, i.e., M is upper anti-triangular.

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(\underbrace{x^{(1)}_{i_1}}_{=x} + \underbrace{x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d}}_{=y}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 0, 1.$$

Recall Oseledets' constructive method

1. $p(x+y) = \sum_{i=0}^{m} p_i (x+y)^i = \sum_{i=0}^{m} p_i \sum_{\alpha=0}^{i} C_i^\alpha x^\alpha y^{i-\alpha} = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M(\alpha,\beta) x^\alpha y^\beta = \underbrace{[1, x, \ldots, x^m]}_{=X_{\leq m}(x)} M \begin{bmatrix} 1 \\ y \\ \vdots \\ y^m \end{bmatrix}$
   where $M(\alpha,\beta) = p_{\alpha+\beta} C^\alpha_{\alpha+\beta}$ for $\alpha + \beta \leq m$ otherwise 0, i.e., M is upper anti-triangular.
2. $p(x+y) = X_{\leq m}(x)^\top M X_{\leq m}(y)$
   $\implies p(x^{(1)}_{i_1} + x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d}) = \mathbf{P}(i_1, \ldots, i_d) = \underbrace{X_{\leq m}(x^{(1)}_{i_1})^\top M}_{=\mathbf{G}_1(i_1) \in \mathbb{R}^{1\times(m+1)}} X_{\leq m}(x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d}).$

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(x^{(1)}_{i_1} + x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 0, 1.$$

Recall Oseledets' constructive method

1. $p(x+y) = \sum_{i=0}^{m} p_i (x+y)^i = \sum_{i=0}^{m} p_i \sum_{\alpha=0}^{i} C_i^{\alpha} x^{\alpha} y^{i-\alpha} = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M(\alpha,\beta) x^{\alpha} y^{\beta} = \underbrace{[1, x, \ldots, x^m]}_{=X_{\leq m}(x)} M \begin{bmatrix} 1 \\ y \\ \vdots \\ y^m \end{bmatrix}$
   where $M(\alpha,\beta) = p_{\alpha+\beta} C_{\alpha+\beta}^{\alpha}$ for $\alpha + \beta \leq m$ otherwise 0, i.e., M is upper anti-triangular.
2. $p(x+y) = X_{\leq m}(x)^{\top} M X_{\leq m}(y)$
   $\Longrightarrow p(x^{(1)}_{i_1} + x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d}) = \mathbf{P}(i_1, \ldots, i_d) = \underbrace{X_{\leq m}(x^{(1)}_{i_1})^{\top} M}_{=\mathbf{G}_1(i_1) \in \mathbb{R}^{1\times(m+1)}} X_{\leq m}(x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d})$.
3. $\mathbf{G}_2(i_2)$ is constructed such that $X_{\leq m}(x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d}) = \mathbf{G}_2(i_2) X_{\leq m}(x^{(3)}_{i_3} + \cdots + x^{(d)}_{i_d})$ holds

$$\Longrightarrow \mathbf{P}(i_1, i_2, \cdots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2) X_{\leq m}(x^{(3)}_{i_3} + \cdots + x^{(d)}_{i_d}).$$

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(x^{(1)}_{i_1} + x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 0, 1.$$

Recall Oseledets' constructive method

1. $p(x+y) = \sum_{i=0}^{m} p_i (x+y)^i = \sum_{i=0}^{m} p_i \sum_{\alpha=0}^{i} C_i^\alpha x^\alpha y^{i-\alpha} = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M(\alpha,\beta) x^\alpha y^\beta = \underbrace{[1, x, \ldots, x^m]}_{=X_{\leq m}(x)} M \begin{bmatrix} 1 \\ y \\ \vdots \\ y^m \end{bmatrix}$
   where $M(\alpha,\beta) = p_{\alpha+\beta} C^\alpha_{\alpha+\beta}$ for $\alpha+\beta \leq m$ otherwise 0, i.e., M is upper anti-triangular.
2. $p(x+y) = X_{\leq m}(x)^\top M X_{\leq m}(y)$
   $\implies p(x^{(1)}_{i_1} + x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d}) = \mathbf{P}(i_1, \ldots, i_d) = \underbrace{X_{\leq m}(x^{(1)}_{i_1})^\top M}_{=\mathbf{G}_1(i_1) \in \mathbb{R}^{1\times(m+1)}} X_{\leq m}(x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d}).$
3. Other cores $\mathbf{G}_2(i_2), \cdots, \mathbf{G}_d(i_d)$ are constructed similarly.

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 0, 1.$$

Recall Oseledets' constructive method

1. $p(x+y) = \sum_{i=0}^{m} p_i (x+y)^i = \sum_{i=0}^{m} p_i \sum_{\alpha=0}^{i} C_i^\alpha x^\alpha y^{i-\alpha} = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M(\alpha,\beta) x^\alpha y^\beta = \underbrace{[1, x, \ldots, x^m]}_{=X_{\leq m}(x)} M \begin{bmatrix} 1 \\ y \\ \vdots \\ y^m \end{bmatrix}$
   where $M(\alpha, \beta) = p_{\alpha+\beta} C_{\alpha+\beta}^{\alpha}$ for $\alpha + \beta \leq m$ otherwise 0, i.e., M is upper anti-triangular.
2. $\implies p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \underbrace{X_{\leq m}(x_{i_1}^{(1)})^\top M}_{=\mathbf{G}_1(i_1) \in \mathbb{R}^{1\times(m+1)}} X_{\leq m}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}).$
3. Other cores $\mathbf{G}_2(i_2), \cdots, \mathbf{G}_d(i_d)$ are constructed similarly.

Disadvantage: Unstable since the cores contain large binomial coefficients & powers e.g. $(x_{i_1}^{(1)})^\alpha$.

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(x^{(1)}_{i_1} + x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 0, 1.$$

Recall Oseledets' constructive method

1. $p(x+y) = \sum_{i=0}^{m} p_i (x+y)^i = \sum_{i=0}^{m} p_i \sum_{\alpha=0}^{i} C_i^\alpha x^\alpha y^{i-\alpha} = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M(\alpha,\beta) x^\alpha y^\beta = \underbrace{[1, x, \ldots, x^m]}_{=X_{\leq m}(x)} M \begin{bmatrix} 1 \\ y \\ \vdots \\ y^m \end{bmatrix}$
   where $M(\alpha,\beta) = p_{\alpha+\beta} C^\alpha_{\alpha+\beta}$ for $\alpha+\beta \leq m$ otherwise 0, i.e., M is upper anti-triangular.
2. $\implies p(x^{(1)}_{i_1} + x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d}) = \mathbf{P}(i_1, \ldots, i_d) = \underbrace{X_{\leq m}(x^{(1)}_{i_1})^\top M}_{=\mathbf{G}_1(i_1) \in \mathbb{R}^{1\times(m+1)}} X_{\leq m}(x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d})$.
3. Other cores $\mathbf{G}_2(i_2), \cdots, \mathbf{G}_d(i_d)$ are constructed similarly.

Disadvantage: Unstable since the cores contain large binomial coefficients & powers e.g. $(x^{(1)}_{i_1})^\alpha$.

Fix: replace x^α, y^β by Chebyshev poly.: $p(x+y) = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M_m^{\mathrm{cheb}}(\alpha,\beta) T_\alpha(x) T_\beta(y)$.

Direct construction in Chebyshev basis

Fix: Replace monomial expansion

$$p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M(\alpha, \beta) x^{\alpha} y^{\beta} \tag{2}$$

by Chebyshev expansion

$$p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M_m^{\text{cheb}}(\alpha, \beta) T_{\alpha}(x) T_{\beta}(y). \tag{3}$$

Direct construction in Chebyshev basis

Fix: Replace monomial expansion

$$p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M(\alpha,\beta) x^\alpha y^\beta \tag{2}$$

by Chebyshev expansion

$$p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M_m^{\mathrm{cheb}}(\alpha,\beta) T_\alpha(x) T_\beta(y). \tag{3}$$

- **Subtle but crucial difference**:
  - (2) holds for all $x, y \in \mathbb{R}$ for fixed M
  - (3) only holds for $x \in I_x$ and $y \in I_y$ with some intervals I_x, I_y and M_m^{cheb} depend accordingly on these intervals.
  - Algorithmically: carefully track the intervals I_x, I_y the auxiliary variables $x_{i_\ell}^{(\ell)}$ live in during the construction.

Direct construction in Chebyshev basis

Fix: Replace monomial expansion

$$p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M(\alpha,\beta) x^\alpha y^\beta \tag{2}$$

by Chebyshev expansion

$$p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M_m^{\text{cheb}}(\alpha,\beta) T_\alpha(x) T_\beta(y). \tag{3}$$

- **Subtle but crucial difference**:
  - (2) holds for all $x, y \in \mathbb{R}$ for fixed M
  - (3) only holds for $x \in I_x$ and $y \in I_y$ with some intervals I_x, I_y and M_m^{cheb} depend accordingly on these intervals.
  - Algorithmically: carefully track the intervals I_x, I_y the auxiliary variables $x_{i_\ell}^{(\ell)}$ live in during the construction.
- **Rank-adaptivity**: apply truncated SVDs on the fly.

Rank-adaptive construction (Error control)

- All SVD tolerances during the process are chosen as ε.
- We have guaranteed element-wise error control.

Theorem [Benner, Khoromskij, S., in preparation, 2026]

For the QTT approximation of the polynomial $p(x)$ constructed by the rank-adaptive method, we have

$$|\mathbf{P} - \hat{\mathbf{P}}| \leq (\sqrt{m+1} + 1)(d-1)\varepsilon$$

Organization

1. Introduction
2. QTT approximation of polynomials
3. Fully discrete format
4. Numerical experiments
5. Conclusion

ChebTuck to Grid-based tensor

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1, x_2, x_3) \approx f_{\mathbf{m}}(x_1, x_2, x_3) = \sum_{i_1=1}^{R} \sum_{i_2=1}^{R} \sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell)$.

ChebTuck to Grid-based tensor

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

  with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell)$.

- **Goal**: Transform ChebTuck format to a discrete tensor format on a fine grid $n \gg m$:

$$\mathbf{F}_{\mathbf{m}}(i_1,i_2,i_3) := f_{\mathbf{m}}(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}), \quad \{t_{i_\ell}^{(\ell)}\}_{i_\ell=1}^{n} \text{ is a uniform grid in } [-1,1].$$

ChebTuck to Grid-based tensor

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

  with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell)$.
- **Goal**: Transform ChebTuck format to a discrete tensor format on a fine grid $n \gg m$:

$$\mathbf{F}_{\mathbf{m}}(i_1,i_2,i_3) := f_{\mathbf{m}}(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}), \quad \{t_{i_\ell}^{(\ell)}\}_{i_\ell=1}^{n} \text{ is a uniform grid in } [-1,1].$$

- Apply our QTT approximation algorithm to each polynomial $v_{j_\ell}^{(\ell)}(x_\ell)$.

From ChebTuck to QTT-Tucker

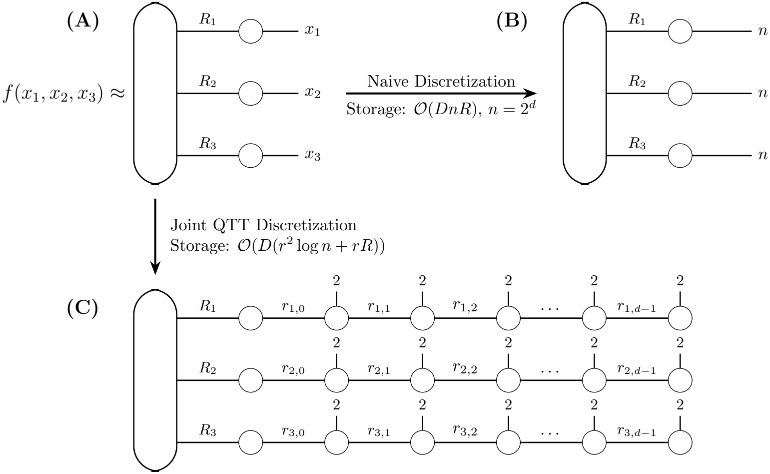


Figure: (A) Continuous ChebTuck. (B) Discrete Tucker ($\mathcal{O}(n)$ storage). (C) QTT-Tucker-like ($\mathcal{O}(\log n)$ storage).

Organization

Random polynomials in monomial basis

- Non-trivial, but practical and not pathological (not too oscillatory) polynomials.
- Polynomials of degree m (up to 300) with random coefficients:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m.$$

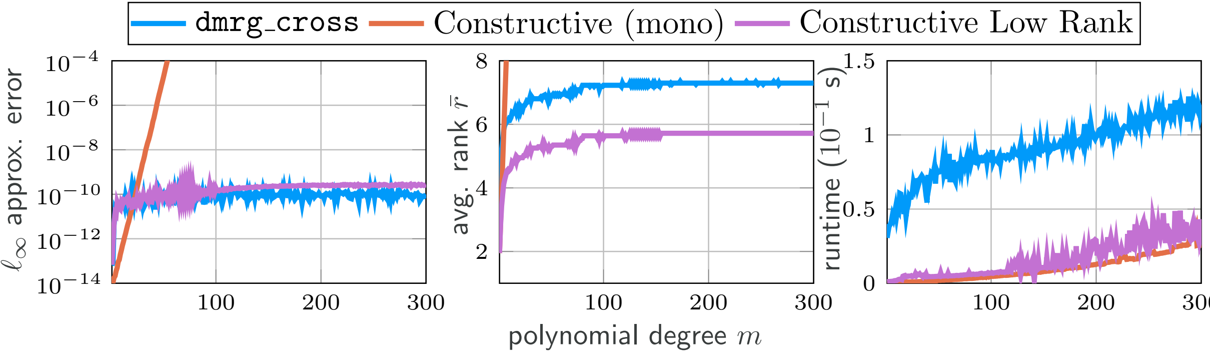
- a_k decays exponentially with random relative noise.
- Discretization on uniform grid with $n = 2^{20}$ points (20 dimensional QTT tensor).
- Tolerances of all SVDs in our method are set to 10^{-12}.
- `dmrg_cross` tolerance is set to 10^{-12}.

Random polynomials in monomial basis

- Polynomials of degree m (up to 300) with random coefficients:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m.$$

- a_k decays exponentially with random relative noise.

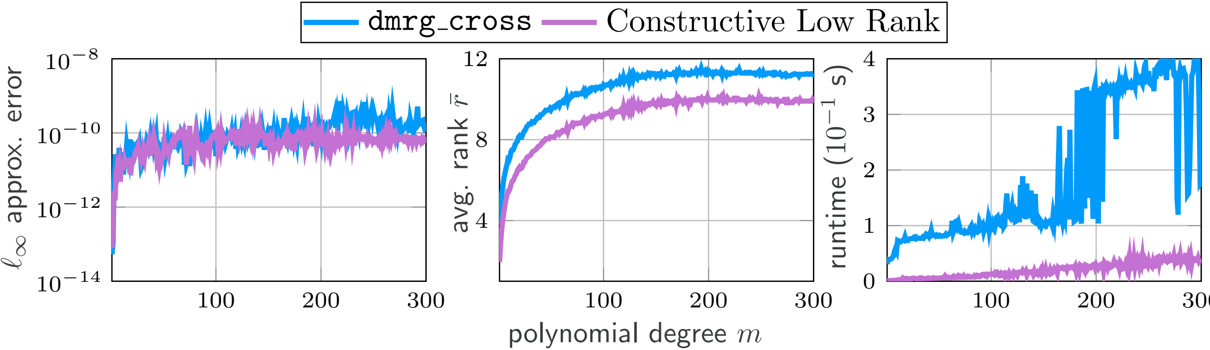


Random polynomials in Chebyshev basis

- Chebyshev polynomials of degree m (up to 300) with random coefficients:

$$p(x) = a_0 + a_1 T_1(x) + a_2 T_2(x) + \cdots + a_m T_m(x).$$

- a_k decays exponentially with random relative noise.



Single Chebyshev polynomials

- A single Chebyshev polynomial $T_m(x)$ of degree m (up to 300).
- Very challenging since it is highly oscillatory.

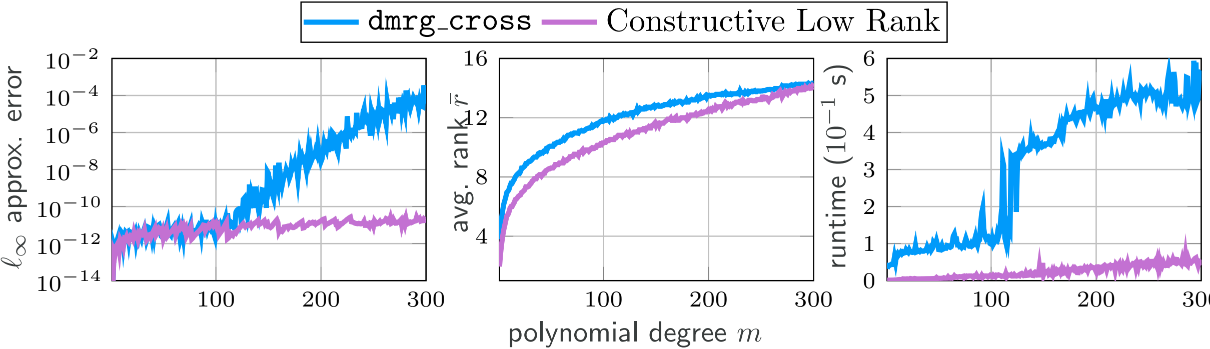


Figure: TTCross vs. Constructive Low Rank.

Test functions

We consider the following 3 test functions $f : [-1, 1]^3 \to \mathbb{R}$:

1. **Biomolecule potential** f_1: the multi-particle potential of the protein Fasciculin 1 comprising 1,228 atoms[2].
2. **Runge function** f_2: the classical 3-dimensional Runge function[3] given by

$$f_2(x, y, z) = \frac{1}{1 + 25(x^2 + y^2 + z^2)}. \tag{4}$$

3. **Wagon function** f_3: the SIAM 100-Dollar, 100-Digit Challenge function[4] defined by

$$\begin{aligned} f_3(x, y, z) = {} & e^{\sin(50x)} + \sin(60e^y)\sin(60z) + \sin(70\sin(x))\cos(10z) \\ & + \sin(\sin(80y)) - \sin(10(x+z)) + \frac{x^2 + y^2 + z^2}{4}. \end{aligned} \tag{5}$$

---

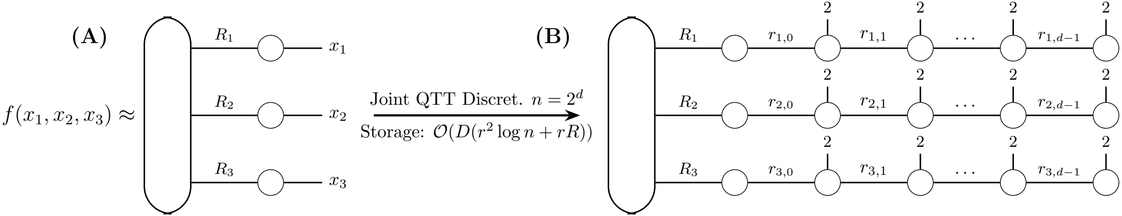
[2] M.H. Le Du, P. Marchot, P.E. Bougis, and J.C. Fontecilla-Camps, J. Biol. Chem., 267:22122–30, 1992.
[3] B. Hashemi and L. N. Trefethen, SIAM J. Sci. Comput., 39(5):C341–C363, 2017.
[4] F. Bornemann, D. Laurie, S. Wagon, and J. Waldvogel. SIAM, 2004.

- Combining ChebTuck

$$f(x_1, x_2, x_3) \approx \sum_{i_1=1}^{R} \sum_{i_2=1}^{R} \sum_{i_3=1}^{R} \beta_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

where $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell)$ with QTT:

$$\mathbf{F}(i_1, i_2, i_3) \approx \boldsymbol{\beta} \times_1 \mathbf{U}^{(1)}(i_1) \times_2 \mathbf{U}^{(2)}(i_2) \times_3 \mathbf{U}^{(3)}(i_3).$$



- Combining ChebTuck

$$f(x_1, x_2, x_3) \approx \sum_{i_1=1}^{R} \sum_{i_2=1}^{R} \sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

where $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell)$ with QTT:

$$\mathbf{F}(i_1, i_2, i_3) \approx \boldsymbol{\beta} \times_1 \mathbf{U}^{(1)}(i_1) \times_2 \mathbf{U}^{(2)}(i_2) \times_3 \mathbf{U}^{(3)}(i_3).$$

Function	**Tucker ranks**	**Polynomial degrees**
f_1	(32, 28, 32)	(129, 129, 129)
f_2	(17, 17, 17)	(189, 189, 189)
f_3	(4, 3, 5)	(662, 1052, 129)

Joint QTT approximation of multiple polynomials

function	method	ℓ_∞ error	storage	avg. rank $\bar{r}$	runtime (10^{-1} s)
f_1	**Constructive Low Rank**	$\mathbf{5.12 \times 10^{-11}}$	**20,550**	21.60	**0.4**
	`dmrg_cross`	7.30×10^{0}	7,770	13.01	5.6
f_2	**Constructive Low Rank**	$\mathbf{1.82 \times 10^{-11}}$	**4,875**	10.49	**0.2**
	`dmrg_cross`	4.15×10^{-8}	5,317	10.98	5.0
f_3	**Constructive Low Rank**	$\mathbf{3.10 \times 10^{-9}}$	**4,182**	9.97	**1.5**
	`dmrg_cross`	4.23×10^{-8}	5,342	11.27	7.1

Table: ChebTuck factors from biomolecule f_1, Runge f_2, and wagon f_3; $n = 2^{20}$.

Fully discrete QTT–Tucker format

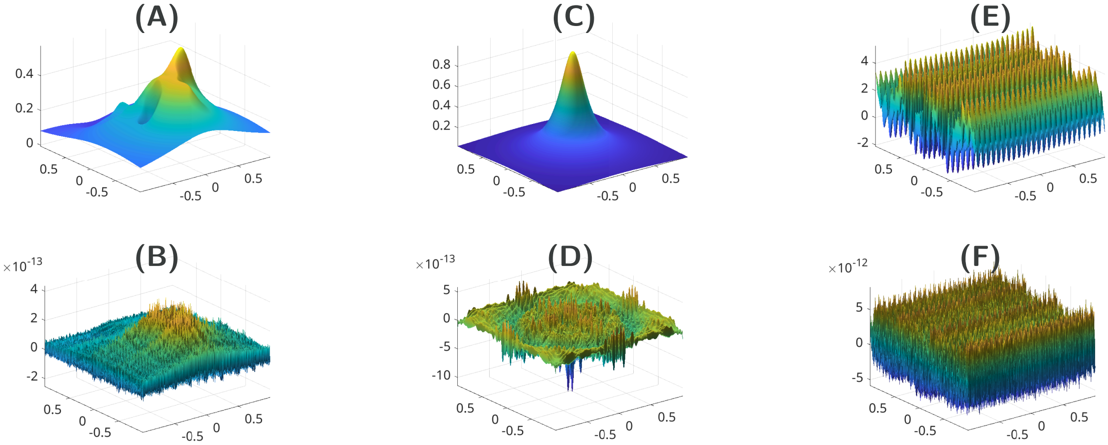


Figure: **(A)**, **(C)**, **(E)**: Surfaces reconstructed from the fully discrete QTT-Tucker format for $f_i(x, y, 0)$.
(B), **(D)**, **(F)**: Corresponding pointwise errors compared to the continuous tensors.

Conclusion

- A framework to bridge continuous and discrete tensor formats via QTT.
- Developed stable and rank-adaptive algorithms for QTT approximation of polynomials.
- Numerical experiments demonstrate stability, efficiency, and storage savings.
- **Outlook**: Extend it to general univariate functions and multivariate ridge functions and more application scenarios.
- For more details and reference:

1. P. Benner, B. Khoromskij, B. Sun. *Bridging continuous and discrete tensor representations of multivariate functions using QTT*, in preparation, 2026.
2. P. Benner, B. Khoromskij, V. Khoromskaia, B. Sun. *A mesh-free hybrid Chebyshev-Tucker tensor format with applications to multi-particle modelling*, arXiv:2505.02319, 2025.

Thank you for your attention!